REMARKS

Claims 1-29 are pending in this application. By this Amendment, the specification and claims 1-2, 4-5, 7-11, 13, and 17-26 are amended, and claims 27-29 are added. No new matter is added. Support for the claims can be found throughout the specification, including the original claims, and the drawings. Reconsideration in view of the above amendments and following remarks is respectfully requested.

The Examiner is thanked for the courtesies extended to Applicant's representative at the January 28, 2005 personal interview. The points discussed are incorporated herein.

Claims 1, 10 and 18 are rejected under 35 U.S.C. §112, second paragraph, for insufficient antecedent basis. Each of the Examiner's comments has been addressed in amending claims 1, 10, and 18. Accordingly, the rejection is obviated and should be withdrawn.

Claims 6, 14, and 23 are rejected under 35 U.S.C. §112, first paragraph, as failing to comply with the enablement requirement. The rejection is respectfully traversed.

The specification has been amended to recite that the Kernal processes a fault generated in the Kernal area as a simple fault. No new matter is added. Accordingly, the rejection is obviated and should be withdrawn.

Claims 1-26 are rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,282,657 to Kaplan et al. (hereinafter "Kaplan). The rejection is respectfully traversed.

Independent claims 1 and 10 each recite a method for verifying user memory validity in an OS (Operating System). The method of claim 1 includes generating a system call, declaring

areas containing certain functions as a safeguard function area, verifying a validity of a user buffer address area using a user buffer address checking function, if the user buffer address area is not valid, determining whether the user buffer address checking function is a function in the safeguard function area by calling an exception processor, if the user buffer address function is identified as a function in the safeguard function area, identifying a safeguard function area identifier by calling a safeguard exception processor, identifying whether the user buffer address checking function is defined in the system by identifying the safeguard function area identifier, and returning an error value to the user processor if the user buffer address checking function is defined in the system. The method of claim 10 includes performing a system call for a user process, declaring a validity checking function as a safeguard function in a safeguard function area, determining whether a user memory area is valid using the validity checking function, if the user memory area is not valid, identifying whether the validity checking function is declared as the safeguard function by calling an exception processor, if the validity checking function is in the safeguard function area, calling a safeguard exception processor, identifying an identifier of the safeguard function area, recognizing via the safeguard function area identifier that a subject of the process is the validity checking function and identifying whether the validity checking function is defined in the system, and if the validity checking function is defined in the system, returning an error to the user processor.

Further, independent claim 18 recites a computer-readable medium having stored thereon a sequence of instructions which, when executed by a processor, cause the processor to at least

perform a method. The method includes generating a system call for a user process, declaring areas containing certain functions a safeguard area, verifying validity of a user buffer address area using a user buffer address checking function, if the user buffer address area is not valid, determining whether the user buffer address checking function is declared as a function in the safeguard function area by calling an exception processor, if the user buffer address checking function is identified as a function in the safeguard area, identifying a safeguard identifier of the safeguard function area by calling a safeguard exception processor, confirming whether the user buffer address checking function is defined in the system by identifying the safeguard function area identifier, and returning an error value to the user processor if the user buffer address checking function is defined in the system.

In contrast, Kaplan discloses a kernel mode protection method and apparatus. The protection circuit disclosed by Kaplan operates in two modes: user or kernel. A processor is reset if a security violation has occurred, such as by attempting to access protected Kernel memory in user mode. A program fetch supervisory circuit compares addresses to a predetermined address to determine if a security violation has occurred and controls a flip-flop current which switches between the two modes. A data fetch supervisor circuit compares data addresses to a protected memory address range. A security violation occurs if the processor is in user mode and the data address is in protected memory, which resets the processor.

However, as discussed at the personal interview, Kaplan does not disclose or suggest, referring to independent claims 1 and 18, declaring areas containing certain functions as a

safeguard function area. Similarly, with respect to independent claim 10, Kaplan does not disclose or suggest declaring a validity checking function as a safeguard function in a safeguard function area. Kaplan merely discloses that a certain range of addresses in the data memory 22 are protected, that is, protected registers and RAM 24. This range of addresses is protected at all times. See col. 2, lines 12-15 of Kaplan. Protecting an address or range of addresses so as to be accessible only to certain secure Kernal firmware is not the same as safeguarding functions as a safeguard function area, in particular in response to a system call.

Additionally, as further discussed at the personal interview, Kaplan does not disclose or suggest, with respect to independent claims 1 and 18, verifying a validity of a user buffer address area using a user buffer address checking function. Similarly, with respect to independent claim 10, Kaplan does not disclose or suggest determining whether a user memory area is valid using the validity checking function. Rather, Kaplan merely discloses that the Kernal data fetch supervisor circuit 20 compares the data memory address fetch to the address range of the protected registers and RAM 24. There is no disclosure of suggestion that Kaplan verifies the validity of the data memory address fetch.

Additionally, Kaplan does not disclose or suggest, if the user buffer address area is not valid, determining whether the user buffer address checking function is a function in the safeguard function area by calling an exception processor; if the user buffer address checking function is identified as a function in the safeguard function area, identifying a safeguard function area identifier by calling a safeguard exception processor; identifying whether the user

buffer address checking function is defined in the system by identifying the safeguard function area identifier; or returning an error value to the user processor if the user buffer address checking function is defined in the system with respect to independent claims 1 and 18. Similarly, Kaplan does not disclose or suggest, with respect to independent claim 10, if the user memory area is not valid, identifying whether the validity checking function is declared as the safeguard function by calling an exception processor; if the validity checking function is in the safeguard function area, calling a safeguard exception processor; identifying an identifier of the safeguard function area; recognizing via the safeguard function area identifier that a subject of the process is the validity checking function and identifying whether the validity checking function is defined in the system; and if the validity checking function is defined in the system, returning an error to the user processor.

Accordingly, the rejection of independent claims 1, 10, and 18 over Kaplan should be withdrawn. Dependent claims 2-9, 11-17, and 19-26 are allowable at least for the reasons discussed above with respect to independent claims 1, 10, and 18, from which they respectfully depend, as well as for their added features.

In view of the foregoing amendments and remarks, it is respectfully submitted that the application is in condition for allowance. If the Examiner believes that any additional changes would place the application in better condition for allowance, the Examiner is invited to contact the undersigned attorney, **Carol L. Druzbick**, at the telephone number listed below.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this, concurrent and future replies, including extension of time fees, to Deposit Account 16-0607 and please credit any excess fees to such deposit account.

Respectfully submitted,
FLESHNER & KIM, LLP

Carol L. Druzbick
Registration No. 40,287

P.O. Box 221200
Chantilly, Virginia 20153-1200
(703) 766-3701 CLD/kah
Date: February 16, 2005

Please direct all correspondence to Customer Number 34610